**Soluciones:**

**Ejer 1 A**:

4 11

4 20

4 22

6 20

6 22

8 22

**B**:

3

2

1

0

**Ejer 2**:

def cubo(n):

return n\*\*3

def suma\_cubo(a, b, c):

return(cubo(a)+cubo(b)+cubo(c))

def cubos(n):

x=n//100

y=n%10

z=(n//10)%10

if(suma\_cubo(x,y,z)==n):

return True

else:

return False

for i in range(100,1000):

if(cubos(i)):

print(i)

**Ejer 3:**

def busca (n,lista):

    for i in range (len(lista)):

        if lista[i]==n:

            return (n)

    return -1

def situacion (saldo):

    informe=""

    if saldo>=0:

        informe="Saldo positivo  o cero"

    elif saldo>-2000 and saldo<0:

        informe="Descubierto Permitodo"

    elif saldo<-2000:

        informe=-1

    return informe

# iNGRESO DE DATOS

clientes=["A","B","C","D"]

saldo=[3000,0,-6000,-1000]

tel=[151,152,153,154]

nom=input("ingresar Nombre")

j=busca(nom,clientes)

if j!=-1:

    j=busca(nom,clientes)

    print ("Cliente encontrado")

    print ("saldo del cliente ",saldo[j])

    if situacion (saldo[j])==-1:

        print ("Llamar al tel", tel[j])

    else:

        print (situacion (saldo[j]))

else:

    print ("Cliente NO ENCONTRADO")

**Ejer 4:**

def farmaciasDelVisitador(v):

lista = []

encontrado = False

i = 0

vismed = visitadoresMedicos()

cant = len(vismed)

while (i < cant and not encontrado):

if vismed[i] == v:

encontrado = True

i = i + 1

if encontrado:

loc1 = LocalidadVisitador(v)

for y in farmacias():

loc2 = LocalidadFarmacia(y)

if Distancia(loc1, loc2) <= 40:

lista.append(y)

return lista

**Ejercicio 1: (2 puntos)**

Dar la salida de los siguientes programas:

|  |  |
| --- | --- |
| **(a)**  n=[4,6,8,10]  m=[7,8,0,2,11,20,22]  i=n[m[2]]  j=i  while(i<n[3]):  j=i  while(j<m[0]):  print(n[i-4], m[j])  j=j+1  i=i+1 | **(b)**  l=[3,2,1,0,-1]  for i in range(len(str(l[4]))+2):  print (l[i]) |

**Ejercicio 2: (2.5 puntos)**

Desarrollar las funciones y el programa principal tales que determine cuáles son los **números de tres cifras** (que no comiencen con cero) que cumplan que la suma del cubo de cada una de sus cifras individuales sea igual al número propuesto. Es decir aquellos que cumplan: .Siendo ABC un numero de tres cifras y no un producto.

Se pide: hacer tres funciones y el programa principal con su código

**Cubo(n)** recibe un número entero de una cifra y devuelve su cubo.

**SumaCubos(a, b, c)** recibe tres enteros de una cifra y devuelve la suma del cubo de los tres, utilizando la función anterior.

**Cubos(n)** recibe un entero de tres cifras y devuelve True en caso que se cumpla que dicho número es igual a la suma del cubo de sus cifras individuales y retorne False en caso contrario. Usar la función anterior.

**Ej**: para el número , se cumple la condición, no así para el numero 120.

**Ejercicio 3: (2.5 puntos)**

Dada una lista de Apellidos de Clientes **no repetidos** que se corresponde con otra saldo y con otra con sus teléfonos.

Realizar un sistema **implementando funciones necesarias** para que al ingresar el **apellido** de un cliente responda a la siguiente tabla con las acciones que se describen:

* Si el saldo es mayor o igual a 0 informe: **SIN DEUDA**
* Si el saldo es menor que 0 pero mayor o igual a -2000 informe: **DESCUBIERTO PAUTADO**,
* Si el monto es menor a -2000 informe **DEBE REGULARIZAR** , **mostrar su saldo** **y el teléfono para llamarlo**

Si el apellido no se encuentra en la lista debe informar**: No Existe Cliente con ese Apellido**

Por ejemplo:

**LisCli** = [“AlvarezC”,”**AlvarezW**”,” “**Bieira**”, “Datola”, “Flamingo”, “**Irusta**”] Cadenas de caracteres

**LisSaldo** = [-50, **0**, **-150000**, -30000, 45000,**-1500**] Enteros

**LisTel** = [1520004000, 1570701111, **1580801010**, 1544445555, 1577779999,1533334545] Cadenas

Si ingresamos:

* **Bieira** el sistema debe informar : DEBE REGULARIZAR, Saldo **-150000**, teléfono **1580801010**
* Cataldo debe informar **No Existe Cliente con ese Apellido**
* **AlvarezW** informará **SIN DEUDA**
* **Irusta** informará **DESCUBIERTO PAUTADO**

**Ejercicio 4: (3 puntos)**

Se desea hacer un programa para que los visitadores médicos de la provincia de Bs. As. visiten determinadas farmacias de esa provincia y así ofrecer los nuevos productos. Las farmacias que deberá visitar cada visitador médico son las que estén a una distancia menor o igual a 40 Km del lugar de su residencia.

Se cuentan con varias funciones a saber:

**Farmacias()** : devuelve una lista de todas las farmacias de la provincia de Bs. As. que deberán ser visitadas.

**VisitadoresMedicos():** devuelve una lista con los códigos únicos de cada visitador médico de esa provincia.

**LocalidadVisitador(v):** devuelva la localidad de residencia del visitador v.

**LocalidadFarmacia(f):** devuelve la localidad de la farmacia f.

**Distancia(loc1, loc2) :** devuelve la distancia en Km entre las localidades loc1 y loc2.

Para resolver lo planteado se deberá construir una función tal que para cada visitador, devuelva la lista de

farmacias que deberá visitar.

Desestimamos que ocurra que algunas farmacias resulten ser visitadas por más de un visitador…